

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2003-332978  
(P2003-332978A)

(43) 公開日 平成15年11月21日 (2003.11.21)

(51) Int.Cl.<sup>7</sup>  
H 04 B 7/26  
G 06 F 12/14  
H 04 Q 7/38

識別記号  
3 1 0

F I  
G 06 F 12/14  
H 04 B 7/26

テマコード(参考)  
3 1 0 A 5 B 0 1 7  
Z 5 K 0 6 7  
1 0 9 R

審査請求 未請求 請求項の数17 OL (全 17 頁)

(21) 出願番号

特願2002-143608(P2002-143608)

(22) 出願日

平成14年5月17日 (2002.5.17)

(71) 出願人 392026693

株式会社エヌ・ティ・ティ・ドコモ  
東京都千代田区永田町二丁目11番1号

(71) 出願人 500426478

株式会社ケイ・ラボラトリ  
東京都港区虎ノ門4-3-20 神谷町森ビル2F

(72) 発明者 神谷 大

東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内

(74) 代理人 100098084

弁理士 川▲崎▼ 研二 (外1名)

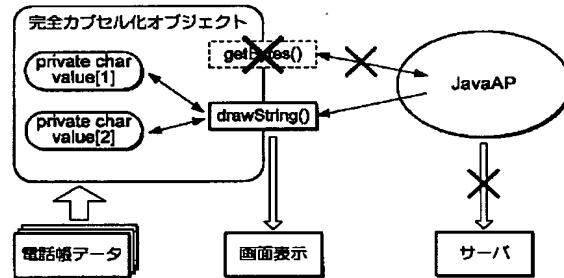
最終頁に続く

(54) 【発明の名称】 通信装置、プログラムおよび記録媒体

(57) 【要約】

【課題】 ネットワークを介して提供されるプログラムの利便性を損なわせることなく、かつ、このようなプログラムに対するセキュリティを確保すること。

【解決手段】 携帯電話機40は、当該携帯電話機40に記憶されている重要度の高いデータや著作権を保護しなければならないコンテンツなどを、これらのデータやコンテンツなどをカプセル化して有するとともに、当該カプセル化されたデータを操作元のプログラムへ引き渡すことのないメソッドのみを有する完全カプセル化オブジェクトとして扱う。携帯電話機40にダウンロードされたJavaAPは、完全カプセル化オブジェクトに対し、このオブジェクトに備わるメソッドを用いて当該オブジェクト内のカプセル化されたデータに対する操作を指令する。



【特許請求の範囲】

【請求項 1】 プログラムを受信する受信手段と、当該通信装置に記憶されているデータのうち、前記受信手段により受信されたプログラムを実行した場合に使用されるデータを特定する特定手段と、前記特定手段により特定されたデータをカプセル化して有するオブジェクトであって、カプセル化されたデータを当該オブジェクトの外部から操作するためのメソッドのうち、前記データを操作元のプログラムへ引き渡すことのないメソッドのみを有するオブジェクトを生成する生成手段と、前記受信手段により受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、当該通信装置に記憶されているデータのうち前記特定手段により特定されたデータへのアクセスを禁止する一方、前記生成手段により生成されたオブジェクトへのアクセスを許可するアクセス制御手段とを有することを特徴とする通信装置。

【請求項 2】 プログラムを受信する受信手段と、当該通信装置に記憶されているデータのうち、前記受信手段により受信されたプログラムを実行した場合に使用されるデータを特定する特定手段と、前記特定手段により特定されたデータに対して予め設定されている当該データの保護の要否を示す保護要否情報を基づいて、カプセル化された前記データと、当該カプセル化されたデータを操作するためのメソッドとを有するオブジェクトを、前記データを操作元のプログラムへ引き渡すメソッドを有する第 1 のオブジェクトとするのか、前記データを操作元のプログラムへ引き渡すメソッドを有していない第 2 のオブジェクトとするのかを決定する決定手段と、前記決定手段により決定された第 1 または第 2 のオブジェクトを生成する生成手段と、前記受信手段により受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、当該通信装置に記憶されているデータのうち前記特定手段により特定されたデータへのアクセスを禁止する一方、前記生成手段により生成されたオブジェクトへのアクセスを許可するアクセス制御手段とを有することを特徴とする通信装置。

【請求項 3】 前記決定手段は、前記特定手段により特定されたデータに対して設定されている前記保護要否情報が前記データの保護要示している場合に、前記生成手段により生成されるオブジェクトを前記第 2 のオブジェクトに決定する一方、前記保護要否情報が前記データの保護否を示している場合に、前記生成手段により生成されるオブジェクトを前記第 1 のオブジェクトに決定することを特徴とする請求項 2 に記載の通信装置。

【請求項 4】 前記受信手段により受信されるプログラムは、前記生成手段により生成されたオブジェクトに対

し、このオブジェクトが有するメソッドを用いて当該オブジェクト内のカプセル化されたデータに対する操作を指令するように作成されていることを特徴とする請求項 1 または 2 に記載の通信装置。

【請求項 5】 前記特定手段は、前記受信手段により受信されたプログラムを実行する場合に、当該通信装置に記憶されているデータのうち、前記プログラムを実行した場合に使用されるデータを特定することを特徴とする請求項 1 または 2 に記載の通信装置。

【請求項 6】 前記生成手段により生成されたオブジェクトを前記プログラムの実行終了に応じて削除する削除手段をさらに有することを特徴とする請求項 5 に記載の通信装置。

【請求項 7】 当該通信装置に記憶されているデータのうち、重要度が高いデータに対して当該データの保護要示す前記保護要否情報を設定する一方、重要度が低いデータに対して当該データの保護否を示す前記保護要否情報を設定する設定手段をさらに有することを特徴とする請求項 2 に記載の通信装置。

【請求項 8】 当該通信装置に記憶されているデータのうち、著作権を保護すべきデータに対して当該データの保護要示す前記保護要否情報を設定する一方、著作権が放棄されたデータに対して当該データの保護否を示す前記保護要否情報を設定する設定手段をさらに有することを特徴とする請求項 2 に記載の通信装置。

【請求項 9】 前記保護要否情報を操作入力に応じて設定する設定手段をさらに有することを特徴とする請求項 2 に記載の通信装置。

【請求項 10】 当該通信装置に記憶されているデータのうち前記特定手段により特定されたデータが更新されると、当該データの更新内容に従って前記生成手段により生成されたオブジェクト内にカプセル化されているデータを更新する更新手段をさらに有することを特徴とする請求項 1 または 2 に記載の通信装置。

【請求項 11】 当該通信装置は、パケット通信機能を有する携帯電話機であることを特徴とする請求項 1 または 2 に記載の通信装置。

【請求項 12】 前記プログラムは、Java プログラミング言語で記述されたプログラムであって、当該通信装置は、前記プログラムを実行するための Java 実行環境を有していることを特徴とする請求項 1 または 2 に記載の通信装置。

【請求項 13】 通信装置に、  
プログラムを受信する受信ステップと、  
前記通信装置に記憶されているデータのうち、前記受信ステップにて受信されたプログラムを実行した場合に使用されるデータを特定する特定ステップと、  
前記特定ステップにて特定されたデータをカプセル化して有するオブジェクトであって、カプセル化されたデータを当該オブジェクトの外部から操作するためのメソッド

ドのうち、前記データを操作元のプログラムへ引き渡すことのないメソッドのみを有するオブジェクトを生成する生成ステップと、

前記受信ステップにて受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、前記通信装置に記憶されているデータのうち前記特定ステップにて特定されたデータへのアクセスを禁止する一方、前記生成ステップにて生成されたオブジェクトへのアクセスを許可するアクセス制御ステップとを実行させるためのプログラム。

【請求項14】通信装置に、

プログラムを受信する受信ステップと、

前記通信装置に記憶されているデータのうち、前記受信ステップにて受信されたプログラムを実行した場合に使用されるデータを特定する特定ステップと、

前記特定ステップにて特定されたデータに対して予め設定されている当該データの保護の要否を示す保護要否情報に基づいて、カプセル化された前記データと、当該カプセル化されたデータを操作するためのメソッドとを有するオブジェクトを、前記データを操作元のプログラムへ引き渡すメソッドを有する第1のオブジェクトとするのか、前記データを操作元のプログラムへ引き渡すメソッドを有していない第2のオブジェクトとするのかを決定する決定ステップと、

前記決定ステップにて決定された第1または第2のオブジェクトを生成する生成ステップと、

前記受信ステップにて受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、当該通信装置に記憶されているデータのうち前記特定ステップにて特定されたデータへのアクセスを禁止する一方、前記生成ステップにて生成されたオブジェクトへのアクセスを許可するアクセス制御手段とを実行させるためのプログラム。

【請求項15】前記決定ステップでは、前記特定ステップにて特定されたデータに対して設定されている前記保護要否情報が前記データの保護要示している場合に、前記生成ステップにて生成されるオブジェクトを前記第2のオブジェクトに決定する一方、前記保護要否情報が前記データの保護否を示している場合に、前記生成ステップにて生成されるオブジェクトを前記第1のオブジェクトに決定することを特徴とする請求項14に記載のプログラム。

【請求項16】前記受信ステップにて受信されるプログラムは、前記生成ステップにて生成されたオブジェクトに対し、このオブジェクトが有するメソッドを用いて当該オブジェクト内のカプセル化されたデータに対する操作を指令するように作成されていることを特徴とする請求項13または14に記載のプログラム。

【請求項17】請求項13ないし16のいずれかの請求項に記載のプログラムを記録したコンピュータ読み取

り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、通信装置のセキュリティを確保するための技術ならびにコンテンツの著作権を保護するための技術に関する。

【0002】

【従来の技術】例えば、パケット通信機能を有する携帯電話機やパーソナルコンピュータなどの通信装置は、インターネットに接続されているサーバから様々なプログラムをダウンロードすることができる。

【0003】ところで、インターネットなどのオープンネットワークでは、世界中の様々な人々が自由に情報の公開やプログラムの提供を行うことができる。オープンネットワークは、このような利点を有する反面、例えば、悪意のある個人や団体が通信装置内に記憶されているデータを密かに盗み出すプログラムを提供していたり、あるいは悪意は無いものの通信装置において動作させると不具合を引き起こしてしまうプログラムが提供されてしまうことがある。

【0004】したがって、ネットワークを介して提供されたプログラムに対して通信装置の内部および外部のリソースを何ら制限することなくアクセスできるようにしてしまうと、例えば、通信装置内に記憶されているユーザーの電話番号やメールアドレス、銀行口座番号などが勝手に読み出され、通信装置の外へ流出してしまうといった事態が生じ得る。

【0005】このため、例えば、Java（登録商標）言語で記述されたプログラムを実行することが可能な通信装置においては、ネットワークを介して提供されたJavaプログラムを実行している場合に、このJavaプログラムの実行過程においてアクセスすることのできるリソースをごく限られたものだけに制限しており、これにより信頼性を完全に保証することのできないプログラムが、例えば、通信装置内のアドレス帳データやユーザーの個人情報などにアクセスすることを禁じている。

【0006】

【発明が解決しようとする課題】上述したアクセス制限の仕組みは、通信装置におけるセキュリティを確保する上で一定の効果を奏するものの、ネットワークを介して提供されるプログラムに対して様々な動作制限を課すことになる。すなわち、このようなアクセス制限は、ネットワークを介して提供されるプログラムが本来有する、通信装置における機能の変更や追加などを自由に行えるという利便性を損なう要因であった。

【0007】しかしながら、ネットワークを介して提供されたプログラムの実行に際し、何らアクセスの制限を行なわない場合、前述した悪意のある個人や団体が提供するプログラムや、通信装置において動作させると不具合を引き起こしてしまうプログラムなどによる被害が、

このプログラムを実行した通信装置のみならず、この通信装置と通信を行った他の電子機器にまで及んでしまうおそれがある。

【0008】本発明は、以上説明した事情に鑑みてなされたものであり、ネットワークを介して提供されるプログラムの利便性を損なわせることなく、かつ、このようなプログラムに対するセキュリティを確保することのできる通信装置、プログラムおよび記録媒体を提供することを目的としている。

【0009】

【課題を解決するための手段】上記課題を解決するために、この発明は、プログラムを受信する受信手段と、当該通信装置に記憶されているデータのうち、前記受信手段により受信されたプログラムを実行した場合に使用されるデータを特定する特定手段と、前記特定手段により特定されたデータをカプセル化して有するオブジェクトであって、カプセル化されたデータを当該オブジェクトの外部から操作するためのメソッドのうち、前記データを操作元のプログラムへ引き渡すことのないメソッドのみを有するオブジェクトを生成する生成手段と、前記受信手段により受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、当該通信装置に記憶されているデータのうち前記特定手段により特定されたデータへのアクセスを禁止する一方、前記生成手段により生成されたオブジェクトへのアクセスを許可するアクセス制御手段とを有する通信装置を提供する。

【0010】また、この発明は、通信装置に、プログラムを受信する受信ステップと、前記通信装置に記憶されているデータのうち、前記受信ステップにて受信されたプログラムを実行した場合に使用されるデータを特定する特定ステップと、前記特定ステップにて特定されたデータをカプセル化して有するオブジェクトであって、カプセル化されたデータを当該オブジェクトの外部から操作するためのメソッドのうち、前記データを操作元のプログラムへ引き渡すことのないメソッドのみを有するオブジェクトを生成する生成ステップと、前記受信ステップにて受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、前記通信装置に記憶されているデータのうち前記特定ステップにて特定されたデータへのアクセスを禁止する一方、前記生成ステップにて生成されたオブジェクトへのアクセスを許可するアクセス制御ステップとを実行させるためのプログラムを提供する。

【0011】この発明によれば、通信装置は、受信したプログラムに基づく処理の実行中に、当該処理において使用されるデータをカプセル化して有しているオブジェクトにアクセスすることが可能となる。まだ、通信装置は、受信したプログラムに基づく処理の実行中に、オブジェクト内にカプセル化されているデータそのものを取

得することはできないが、このオブジェクトに備わるメソッドを用いてオブジェクト内のデータに対する操作を行うことが可能となる。

【0012】また、この発明は、プログラムを受信する受信手段と、当該通信装置に記憶されているデータのうち、前記受信手段により受信されたプログラムを実行した場合に使用されるデータを特定する特定手段と、前記特定手段により特定されたデータに対して予め設定されている当該データの保護の要否を示す保護要否情報を基づいて、カプセル化された前記データと、当該カプセル化されたデータを操作するためのメソッドとを有するオブジェクトを、前記データを操作元のプログラムへ引き渡すメソッドを有していない第1のオブジェクトとするのか、前記データを操作元のプログラムへ引き渡すメソッドを有していない第2のオブジェクトとするのかを決定する決定手段と、前記決定手段により決定された第1または第2のオブジェクトを生成する生成手段と、前記受信手段により受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、当該通信装置に記憶されているデータのうち前記特定手段により特定されたデータへのアクセスを禁止する一方、前記生成手段により生成されたオブジェクトへのアクセスを許可するアクセス制御手段とを有する通信装置を提供する。

【0013】また、この発明は、通信装置に、プログラムを受信する受信ステップと、前記通信装置に記憶されているデータのうち、前記受信ステップにて受信されたプログラムを実行した場合に使用されるデータを特定する特定ステップと、前記特定ステップにて特定されたデータに対して予め設定されている当該データの保護の要否を示す保護要否情報を基づいて、カプセル化された前記データと、当該カプセル化されたデータを操作するためのメソッドとを有するオブジェクトを、前記データを操作元のプログラムへ引き渡すメソッドを有する第1のオブジェクトとするのか、前記データを操作元のプログラムへ引き渡すメソッドを有していない第2のオブジェクトとするのかを決定する決定ステップと、前記決定ステップにて決定された第1または第2のオブジェクトを生成する生成ステップと、前記受信ステップにて受信されたプログラムを実行している場合にアクセスすることのできるリソースを制限し、当該通信装置に記憶されているデータのうち前記特定ステップにて特定されたデータへのアクセスを禁止する一方、前記生成ステップにて生成されたオブジェクトへのアクセスを許可するアクセス制御手段とを実行させるためのプログラムを提供する。

【0014】この発明によれば、通信装置は、受信したプログラムに基づく処理の実行中に、当該処理において使用されるデータをカプセル化して有しているオブジェクトにアクセスすることが可能となる。また、通信装置は、カプセル化するデータに対して設定されている保護

要否情報に基づいて、生成するオブジェクトの種類を決定し、受信したプログラムに基づく処理の実行中に、オブジェクト内にカプセル化されているデータそのものを取得できるようにするか否かを定める。

【0015】

【発明の実施の形態】以下、図面を参照して本発明の実施形態について説明する。なお、各図において共通する部分には、同一の符号が付されている。

【0016】【A-1. 実施形態の構成】

【<1. 通信システムの構成>】図1は、この発明の実施形態に係る通信システム1の構成を例示するブロック図である。同図に示すように通信システム1は、コンテンツサーバ10と、インターネット20と、移動パケット通信網30と、携帯電話機40とを有している。なお、この通信システム1には、本来、多数の携帯電話機40が収容されるが、図面が煩雑になることを防ぐため、図1には、1つの携帯電話機40のみを図示している。また、同様の理由により、図1には、それぞれ1つのコンテンツサーバ10、ゲートウェイサーバ31および基地局32のみを図示している。

【0017】コンテンツサーバ10は、インターネット20および移動パケット通信網30を介して携帯電話機40とパケット通信を行う機能を有している。このコンテンツサーバ10には、携帯電話機40に提供するプログラムや画像データ、楽曲データなどの種々のコンテンツが格納されている。これらのコンテンツの中には、携帯電話機40において実行可能なJavaアプリケーションプログラム（以下、JavaAPと略称する）が格納されている。このJavaAPは、JavaアプレットやJavaアプリケーションなどの、Javaプログラミング言語で記述された携帯電話機40用のアプリケーションプログラムである。

【0018】移動パケット通信網30は、当該移動パケット通信網30に収容される携帯電話機40に対してパケット通信サービスを提供する通信網であり、ゲートウェイサーバ31と基地局32とを有している。なお、通信システム1は、移動パケット通信網30に加え、図示を省略した移動電話網を有している。この移動電話網は、携帯電話機40に対して一般的な移動電話の通話サービスを提供する。

【0019】ゲートウェイサーバ31は、移動パケット通信網30用の通信プロトコルとインターネット20用の通信プロトコルなど、通信プロトコルの異なるデータを相互に変換し、移動パケット通信網30とインターネット20とのデータの授受を中継する。また、基地局32は、移動パケット通信網30の通信サービスエリア内に多数設置されており、自局32がカバーする無線セルに在籍している携帯電話機40と無線通信を行う。

【0020】携帯電話機40は、自機40が在籍している無線セルをカバーする基地局32と無線通信を行う。

また、この携帯電話機40は、移動パケット通信網30およびインターネット20を介してコンテンツサーバ10とパケット通信を行う機能を有しており、コンテンツサーバ10から任意のコンテンツをダウンロードすることができる。

【0021】<2. 携帯電話機の構成>図2は、携帯電話機40のハードウェア構成を例示するブロック図である。同図に示すように携帯電話機40は、無線通信部401と、操作入力部402と、通話処理部403と、通信インターフェース404と、CPU405と、液晶表示部406と、記憶部407とを有しており、これらの各部はバス411により接続されている。

【0022】無線通信部401は、アンテナ401aを備え、基地局32との間で行われる無線通信を制御する。この無線通信部401は、CPU405の制御の下、例えば、送話音声に関するデータやパケット通信用のデータなどを搬送波に重畠して送信信号を生成し、この信号を基地局32へ送信する。また、無線通信部401は、基地局32から送られてくる無線信号をアンテナ401aを介して受信し、この信号を復調して自機40宛の受話音声に関するデータやパケット通信用のデータなどを得る。

【0023】操作入力部402は、数字や文字、操作指示などを入力するための複数のキーを有しており、これらのキーの操作に応じた操作信号をCPU405に出力する。また、通話処理部403は、例えば、マイクロフォンやスピーカ、音声処理部などを有しており、CPU405の制御の下、呼の接続／切断を含む通話処理を行う。

【0024】通信インターフェース404は、通信ケーブルを介して接続された電子機器との有線通信を制御する。なお、この通信インターフェース404は、赤外線通信や、HomeRF（Home Radio Frequency）、Bluetooth（登録商標）などの近距離無線通信を制御するものであってもよい。また、CPU405は、記憶部407に格納されている各種プログラムを実行することにより、バス411を介して接続されている装置各部を制御する。また、液晶表示部406は、液晶表示パネルと、この液晶表示パネルの表示制御を行う駆動回路とを有している。

【0025】記憶部407は、ROM408と、RAM409と、例えば、SRAM（Static-RAM）やEEPROM（Electrically Erasable Programmable-ROM）などの不揮発性メモリ410とを有している。ROM408には、例えば、携帯電話機40用のオペレーティングシステム（以下、OSと略称する）やWeb（WorldWide Web）ブラウザ、Java実行環境を構築するためのソフトウェアなどが記憶されている。また、RAM409は、CPU405のワークエリアとして用いられ、CPU405により実行される各種のプログラムやデータが

一時的に記憶される。

【0026】不揮発性メモリ410には、携帯電話機40の製品出荷時点から当該携帯電話機40に組み込まれているアプリケーションプログラムや、コンテンツサーバ10からダウンロードされたJavaAPなどのコンテンツが格納される。加えて、この不揮発性メモリ410には、電話番号やメールアドレスなどの情報が記憶されているアドレス帳データ、受信あるいは送信した電子メールデータ、着信や発信に関する履歴データ、電子決済を行うためのユーザの銀行口座番号やクレジットカード番号などの各種データが格納される。

【0027】なお、以下、本明細書では、携帯電話機40の製品出荷時点において既にROM408や不揮発性メモリ410に格納されているアプリケーションソフトウェアを、ダウンロードされたJavaAPと区別するため、ネイティブアプリケーションと記載する。このネイティブアプリケーションには、自身がネイティブアプリケーションであることを示す識別情報が付与されている。

【0028】また、不揮発性メモリ410は、型指定テーブル410aと、JARストレージ410bと、個別スクラッチパッド410cと、共通スクラッチパッド410dとを有している。まず、型指定テーブル410aについて図3を参照して説明する。同図に示すように、型指定テーブル410aには、不揮発性メモリ410に格納されている各種のデータのうち、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータなどの、ダウンロードされたJavaAPを実行した場合に当該JavaAPが使用する可能性のあるデータについて、データ名と、当該データを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかを指定する型指定情報とが対応付けられて登録されている。なお、上述したユーザデータとは、携帯電話機40のユーザに関する個人情報であって、例えば、ユーザの氏名や年齢、誕生日、銀行口座番号やクレジットカード番号などである。

【0029】同図に示すように、型指定情報は、“1”または“0”的1ビットデータであって、型指定情報の値を“1”にセットした場合は、オブジェクトの型が「完全カプセル化」型に指定される一方、型指定情報の値を“0”にセットした場合は、オブジェクトの型が「非完全カプセル化」型に指定される。

【0030】なお、カプセル化オブジェクトとは、カプセル化（情報隠蔽）された1以上のデータと、当該カプセル化された各データに対するオブジェクト外部からの操作を可能とするための1以上のメソッドとを有するオブジェクトである。そして、完全カプセル化オブジェクト(tightly encapsulated object)とは、上記カプセル化オブジェクトのうち、オブジェクト内にカプセル化されたデータ自体を当該オブジェクトに対する操作元の

プログラム（例えば、ダウンロードされたJavaAP）へ引き渡すメソッドを一つも持たないように構成したオブジェクトである。また、非完全カプセル化オブジェクトとは、上記カプセル化オブジェクトのうち、オブジェクト内のデータ自体を当該オブジェクトに対する操作元のプログラムへ引き渡すメソッドを少なくとも一つ以上有しているオブジェクトである。完全カプセル化オブジェクトと非完全カプセル化オブジェクトとの差異は、オブジェクト内のカプセル化されたデータ自体を操作元のプログラムへ引き渡すメソッドを有しているか否かである。

【0031】すなわち、図3に示した型指定テーブル410aには、各データ毎に、当該データを完全カプセル化オブジェクトとして扱うのか、それとも非完全カプセル化オブジェクトとして扱うのかが登録されている。例えば、同図において、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータは、携帯電話機40に記憶されているデータの中でも特にセキュリティ上の重要度が高いデータである。一方、携帯電話機40にダウンロードされたJavaAPは、信頼性を完全に保証することのできないプログラムであって、万一、悪意のある第3者が作成した、データを盗み出すJavaAPが携帯電話機40にダウンロードされた場合であっても、このようなJavaAPを介して上述したセキュリティ上の重要度が高いデータが携帯電話機40の外部へ流出してしまうような事態は、極力、防がなければならない。

【0032】したがって、セキュリティ上の重要度が高いデータは完全カプセル化オブジェクトとして扱い、ダウンロードされたJavaAPにデータ自体が引き渡されないようにする必要がある。以上のようなことから、型指定テーブル410aにおいて、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータには、型指定情報の値として“1”（完全カプセル化型）が登録されている。

【0033】また、同図に示した型指定テーブル410aにおいて、コンテンツAおよびコンテンツBは、コンテンツサーバ10からダウンロードされた画像データや音楽データなどのコンテンツである。これらのコンテンツには、コンテンツの提供事業者（以下、コンテンツプロバイダと記載する）により著作権保護フラグが付与されている。ここで、著作権保護フラグは、“1”または“0”的1ビットデータであり、著作権保護フラグの値が“1”にセットされている場合は、このフラグの付与されているコンテンツの著作権を保護しなければならないことを示している。一方、著作権保護フラグの値が“0”にセットされている場合は、このフラグの付与されているコンテンツの著作権が放棄されていることを示している。

【0034】ここで、著作権を保護しなければならない

コンテンツの場合、ダウンロードされたJavaAPにコンテンツデータ自体を引き渡してしまうと、コンテンツプロバイダが許可を与えていないような利用形態でコンテンツがJavaAPにより利用されてしまうおそれや、コンテンツデータがJavaAPを介して携帯電話機40の外部へ不正に転送されてしまうおそれがある。したがって、著作権保護フラグの値として“1”が付与されているコンテンツデータは完全カプセル化オブジェクトとして扱い、ダウンロードされたJavaAPにコンテンツデータ自体が引き渡されないようにする必要がある。以上のようなことから、型指定テーブル410aにおいてコンテンツAには、型指定情報の値として“1”（完全カプセル化型）が登録されている。

【0035】一方、著作権保護フラグの値として“0”が付与されているコンテンツデータは、著作権が放棄されているので、完全カプセル化オブジェクトとして扱う必要がない。この場合、ダウンロードされたJavaAPにコンテンツデータ自体を引き渡しても何ら問題がなく、また、コンテンツデータ自体を引き渡せるようにした方がJavaAPの利便性が高くなる。このため、型指定テーブル410aにおいてコンテンツBには、型指定情報の値として“0”（非完全カプセル化型）が登録されている。

【0036】なお、型指定テーブル410aにおいて、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータには、予め型指定情報の値として“1”がセットされている。また、ダウンロードされたコンテンツについては、コンテンツが携帯電話機40にダウンロードされた際に、このコンテンツに付与されている著作権保護フラグの値に応じた型指定情報の値がCPU405により決定され、コンテンツを識別するコンテンツ識別情報（データ名）とともに型指定テーブル410aに登録される。

【0037】また、携帯電話機40において作成されたデータやユーザがパソコンなどで作成し、通信インタフェース404を介して携帯電話機40に取り込んだ画像データや音楽データなどに対しては、このデータを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかをユーザが操作入力により設定することができる。さらに、型指定テーブル410aの内容を液晶画面に表示して、著作権保護フラグが付与されたコンテンツを除く各データの型指定情報を、ユーザ自身が操作入力により変更可能な構成としてもよい。

【0038】次に、図2に戻り、不揮発性メモリ410は、上述した型指定テーブル410aの他に、JARストレージ410bと、個別スクラッチパッド410cと、共通スクラッチパッド410dとを有している。ここで、JARストレージ410b、個別スクラッチパッド410cおよび共通スクラッチパッド410dについ

て説明する前に、まず、携帯電話機40にダウンロードされるJavaAPについて説明する。JavaAPは、JavaAPの本体プログラムおよび当該本体プログラムの実行に応じて利用される画像ファイルや音声ファイルなどを1つにまとめたJAR（Java Archive）ファイルと、このJARファイルのインストールや起動、ネットワークアクセスなどを制御するための各種制御情報が記述されたADF（Application Descriptor File）とを有している。

【0039】JARストレージ410bおよび個別スクラッチパッド410cには、ダウンロードされたJavaAP毎に当該JavaAP用の記憶領域が設けられる。JARストレージ410b内の各記憶領域には、JavaAPのJARファイルが格納される。また、個別スクラッチパッド410c内の各記憶領域には、例えば、JavaAPがゲームプログラムである場合、今までの得点データやセーブデータなど、JavaAPの利用に応じて発生した当該JavaAP用のデータが格納される。さらに、共通スクラッチパッド410dには、複数のJavaAPが共通して使用するデータが格納される。

【0040】また、ダウンロードの後、JavaAPが携帯電話機40において実行される場合、このJavaAPの実行に伴って携帯電話機40がアクセスすることのできるリソースは、このJavaAPのダウンロード元のコンテンツサーバ10（サイト）と、このJavaAPに対して割り当てられたJARストレージ410bおよび個別スクラッチパッド410c内の記憶領域と、共通スクラッチパッド410dと、のみに制限され、それ以外のリソースにアクセスすることはできない。

【0041】<3. Java実行環境>図4は、携帯電話機40におけるJavaAPの実行環境を説明するための図である。同図に示すように本実施形態に係る携帯電話機40には、JavaAPの実行環境を構築するためのソフトウェアとして、KVM（K Virtual Machine）と、コンフィギュレーションとしてCLDC（Connected Limited Device Configuration）を備えるとともにプロファイルとして通信事業者が独自に策定したオリジナルJava拡張プロファイルを備えたJ2ME（Java 2 Micro Edition）とが組み込まれている。

【0042】KVMは、小型電子機器用に設計変更されたJVM（Java Virtual Machine）であって、JavaAPの実行ファイル形式であるバイトコードをCPU405がOSを介して解釈／実行可能な命令コードに変換する。また、CLDCクラスライブラリは、CLDC用のクラスライブラリである。

【0043】オリジナルJava拡張ライブラリは、CLDCを基礎として携帯電話機に特化した機能を提供するためのクラスライブラリである。このオリジナルJava拡張ライブラリには、例えば、ユーザインタフェースAPI（Application Program Interface）、ネットワーキン

グAPI、スクラッチパッドAPI、完全カプセル化API、非完全カプセル化APIなどが含まれている。

【0044】ここで、ユーザインターフェースAPIは、携帯電話機40のユーザインターフェース機能をサポートするAPIであり、ネットワーキングAPIは、URL(Uniform Resource Locator)により指定されたネットワークリソースへのアクセスをサポートするAPIである。また、スクラッチパッドAPIは、個別スクラッチパッド410cや共通スクラッチパッド410dに対するデータの書き込みや読み出しをサポートするAPIである。さらに、完全カプセル化APIは、完全カプセル化オブジェクトを生成するためのAPIであり、非完全カプセル化APIは、非完全カプセル化オブジェクトを生成するためのAPIである。

【0045】また、携帯電話機40は、CLDCクラスライブラリおよびオリジナルJava拡張ライブラリに加え、メーカ独自拡張ライブラリを有している。このメーカ独自拡張ライブラリは、携帯電話機40を製造する各メーカがそれぞれ独自の機能を提供するためのクラスライブラリである。

【0046】次に、JAM(Java Application Manager)は、OSによる制御の下で、携帯電話機40にダウンロードされたJavaAPや、完全カプセル化オブジェクト、非完全カプセル化オブジェクトなどを管理する機能を有している。例えば、JAMは、JavaAPのインストールや更新、削除を行う機能、不揮発性メモリ410に格納されているJavaAPをリスト表示する機能、JavaAPの実行管理(起動や強制終了など)を行う機能、JavaAPの実行に伴う携帯電話機40のアクセスを制限する機能、完全カプセル化オブジェクトや非完全カプセル化オブジェクトの生成、更新、削除を行なう機能などを有している。

【0047】また、同図に示すように、電話帳機能やブラウザ機能、ネットワーク通信機能などを提供するネイティブアプリケーションは、OSによる制御の下で直接動作する。

【0048】<4. カプセル化オブジェクトの構成>次に、カプセル化オブジェクトについて説明する。図5は、カプセル化オブジェクトについて説明するための模式図である。同図に示すように、カプセル化オブジェクトとは、カプセル化された1以上のデータと、当該カプセル化された各データに対するオブジェクト外部からの操作を可能とするための1以上のメソッドとを有するオブジェクトである。

【0049】同図に示す例では、2つのデータ1,2と、2つのメソッド1,2とを有するカプセル化オブジェクトが例示されている。このカプセル化オブジェクト内のデータ1,2は共にカプセル化されているため、オブジェクトの外部からデータ1,2を直接読み書きすることはできない。したがって、例えば、ダウンロードさ

れたプログラムがカプセル化オブジェクト内のデータ1,2に対してアクセスする場合、プログラムは、メソッド1,2を使用して目的的データ1またはデータ2に対する操作をカプセル化オブジェクトに指令しなければならない。

【0050】ここで、同図に示すメソッド1が、例えば、指定されたデータ 자체を操作元のプログラムへ引き渡すメソッドであれば、操作元のプログラムは、メソッド1を使用してカプセル化オブジェクト内の任意のデータ1,2を取得することが可能である。また、同図に示すメソッド2が、例えば、指定されたデータを液晶画面に表示させるメソッドであれば、操作元のプログラムは、メソッド2を使用してカプセル化オブジェクト内の任意のデータ1,2を画面表示させることができある。ここで注目すべき点は、メソッド2を使用してカプセル化オブジェクト内の任意のデータ1,2を画面表示させたプログラムは、表示させるデータやメソッドを指定してカプセル化オブジェクトへ指令を出すものの、表示させるデータ自体を取得していない点である。

【0051】つまり、データそのものを操作元のプログラムに引き渡すメソッドを1つも有していないカプセル化オブジェクト(完全カプセル化オブジェクト)であれば、操作元のプログラムは、オブジェクト内のデータそのものを取得することはできないが、このオブジェクトに備わるメソッドを使用してオブジェクト内のデータに対する操作を行うことはできる。

【0052】したがって、アドレス帳データや電子メールデータなどを完全カプセル化オブジェクトとして扱うようにすれば、操作元のプログラムがダウンロードされたJavaAPのように信頼性を完全に保証することのできないプログラムであったとしても、当該プログラムにデータ自体を引き渡すことがないので、携帯電話機40におけるセキュリティを確保することができる。また、同時に、このようなプログラムであったとしても、アドレス帳データや電子メールデータなど、従来はセキュリティを確保する観点から一切のアクセスを認めていなかつたデータに対し、完全カプセル化オブジェクトが有するメソッドを用いて操作(アクセス)を行うことができる。

【0053】以上のようなことから本実施形態では、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータなどのセキュリティ上の重要度の高いデータや、著作権を保護しなければならないコンテンツなど、前述した型指定情報の値として“1”がセットされているデータを完全カプセル化オブジェクトとして扱う一方、セキュリティの重要度が低いデータや著作権が放棄されたコンテンツなど、型指定情報の値として“0”がセットされているデータを非完全カプセル化オブジェクトとして扱う。また、ダウンロードされたJavaAPに対して、完全カプセル化オブジェクトや非完全カ

プセル化オブジェクトへのアクセスを許可する。

【0054】図6は、電話帳データに関する非完全カプセル化オブジェクトについて例示する模式図である。なお、本実施形態において、電話帳データはセキュリティ上の重要度が高いため、本来は必ず完全カプセル化オブジェクトとして扱われるデータであるが、ここでは比較のため、あえて非完全カプセル化オブジェクトとして扱った場合について説明する。

【0055】Javaプログラミング言語では、「private」というアクセス修飾子を用いてオブジェクト内のフィールドをprivateフィールドに宣言することで、当該privateフィールドに格納されるデータのカプセル化を図る。つまり、オブジェクト内のフィールドが全てprivateフィールドである場合、各privateフィールドに格納されているデータをオブジェクトの外部から直接読み書きすることができなくなる。このようにした場合、各privateフィールドに格納されているデータに対してオブジェクトの外部からアクセスするには、このオブジェクトに備わるメソッドを使用してデータに対する操作を当該オブジェクトに指令しなければならない。

【0056】同図に示す非完全カプセル化オブジェクトには、2つのprivateフィールドが設けられ、それぞれprivate char value[1]、private char value[2]という電話帳の文字列データが格納されている。また、この非完全カプセル化オブジェクトは、getBytes()、drawString()という2つのメソッドを有している。ここで、getBytes()は、オブジェクト内のデータをバイト配列の形式で操作元のプログラムへ引き渡すメソッドである。したがって、ダウンロードされたJavaAPは、このgetBytes()というメソッドを使用して、非完全カプセル化オブジェクト内の電話帳の文字列データ(private char value[1]、private char value[2])を取得することができる。

【0057】また、drawString()は、オブジェクト内のデータを携帯電話機40の液晶画面に表示させるメソッドである。JavaAPは、このdrawString()というメソッドを使用して、非完全カプセル化オブジェクト内の電話帳の文字列データ(private char value[1]、private char value[2])を液晶画面に表示させることもできる。

【0058】一方、図7は、電話帳データに関する完全カプセル化オブジェクトについて例示する模式図である。同図に示す完全カプセル化オブジェクトが図6に示した非完全カプセル化オブジェクトと異なるのは、完全カプセル化オブジェクトは、上述したgetBytes()のように、オブジェクト内のデータそのものを操作元のプログラムへ引き渡すメソッドを有していない点である。

【0059】すなわち、完全カプセル化オブジェクト

は、カプセル化された上に、オブジェクト内のデータそのものを操作元のプログラムへ引き渡すメソッドを1つも有していない。したがって、ダウンロードされたJavaAPは、drawString()というメソッドを使用してオブジェクト内の電話帳の文字列データ(private char value[1]、private char value[2])を画面表示させることはできるが、電話帳の文字列データそのものを取得することはできない。以上のようなことから、万一、悪意のある第3者が作成した、データを盗み出すJavaAPが携帯電話機40にダウンロードされた場合であっても、このようなJavaAPに電話帳データが引き渡されることはなく、当然、電話帳データがサーバなど携帯電話機40の外部へ送信されることもない。

【0060】ところで、drawString()というメソッドを使用してオブジェクト内の電話帳の文字列データを画面表示させる場合、完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、ネイティブアプリケーションとしてROM408または不揮発性メモリ410に格納されている表示制御プログラムを使用して液晶画面に電話帳の文字列を表示させる。この表示制御プログラムからJavaAPが表示データを取得することができてしまうと、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを用いた意味がなくなってしまう。

【0061】しかしながら、ダウンロードされたJavaAPが実行される場合には、前述したようにJAMのアクセス制限機能により、JavaAPの実行中に携帯電話機40がアクセスすることのできるリソースが制限される。ここで、JavaAPの実行中に携帯電話機40のアクセスが許可されるリソースに表示制御プログラムは含まれていないので、JavaAPが表示制御プログラムから表示データを取得するようなことは一切あり得ない。

【0062】また、カプセル化は、プログラミング言語レベルでのカプセル化と、実行コード(マシン語またはバイトコード)レベルでのカプセル化と考えられる。プログラミング言語レベルでのカプセル化が完全であっても、実行コードレベルでのカプセル化が完全でなければ、データを完全にカプセル化したとは言えない。例えば、プログラミング言語であるC++を用いたプログラムでもprivateフィールドを有するカプセル化オブジェクトを生成することはできる。しかしながら、C++は、単なるプログラミング言語に過ぎないことから、プログラミング言語レベルでのカプセル化しか達成し得ない。

【0063】具体的に説明すると、C++を用いたプログラムにより、オブジェクト内の全てのフィールドをprivateフィールドとして宣言し、カプセル化オブジェクトを生成した場合、確かに、このオブジェクト内のprivateフィールドに格納されているデータを直接読み書きするようなソースコードはコンパイルされることがないので、当然、実行コードが生成されることもない。

【0064】しかしながら、このカプセル化は、コンパ

イラによって保証されているに過ぎず、例えば、悪意のある第3者がコンパイラを改造することで、オブジェクト内のデータを不正に入手することが可能である。また、コンパイラを改造しなくても、悪意のある第3者がハンドアセンブルなどの手段でオブジェクト内のデータを不正に読み出す実行コードを生成するプログラムを作成することも不可能ではない。加えて、ポインタを用いて直接メモリにアクセスしてしまえば、オブジェクト内のデータを入手することができてしまう。

【0065】これに対してJavaの場合、`private`宣言されたフィールドは、`private`属性を有するフィールドであることを示すJavaのバイトコードへコンパイルされる。KVMがクラスファイルをRAM409などへ展開する際も、フィールドの`private`属性は保持されている。したがって、仮にコンパイラを改造してオブジェクト内の`private`フィールドに格納されているデータを不正に読み出すようなバイトコードを生成したとしても、KVMまたはJAMがこれを検知するので、オブジェクト内のデータを入手することはできない。また、Javaはポインタをサポートしていないので、ポインタを用いて直接メモリにアクセスし、オブジェクト内のデータを入手することもできない。

【0066】以上のようなことから、Javaでは、プログラミング言語レベルのみに止まらず、バイトコードレベルでの完全なカプセル化を達成することが可能である。なお、データのカプセル化に際しては、「`private`」の他に「`protected`」や「`package`」などのアクセス修飾子を用いることもできる。以上が本実施形態に係る通信システム1の構成である。

【0067】[A-2. 実施形態の動作] 次に、本実施形態の動作について説明する。なお、携帯電話機40が以下に述べる動作を行う前提として、携帯電話機40は、移動パケット通信網30およびインターネット20を介してコンテンツサーバ10とパケット通信を行い、コンテンツサーバ10からJavaAPをダウンロードして不揮発性メモリ410に格納しているものとする。また、不揮発性メモリ410には、ダウンロードされたJavaAP（コンテンツ）の他に、アドレス帳データや電子メールデータ、ユーザデータなどが格納されており、型指定テーブル410aには、これら各データについて型指定情報が登録されているものとする。

【0068】また、型指定テーブル410aにおいて、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータに対しては、型指定情報の値として“1”がセットされている。また、ダウンロードされたコンテンツに対しては、コンテンツが携帯電話機40にダウンロードされた際に、このコンテンツに付与されている著作権保護フラグの値に応じた型指定情報の値がCPU405により決定され、当該型指定情報およびコンテンツ名が型指定テーブル410aに登録される。

【0069】<1. オブジェクト生成処理>まず、携帯電話機40においてCPU405により実行されるオブジェクト生成処理について図8を参照して説明する。このオブジェクト生成処理は、JAMの機能としてCPU405により実行されるものであり、例えば、画面表示されたプログラムの一覧リストの中から、実行するプログラムが操作入力により指定された場合などに実行される。なお、プログラムの実行を指示する形態は、操作入力によるものに限定されず、例えば、予め定められた時間毎にプログラムの実行が指示される場合や、既に実行されている他のプログラムから実行が指示される場合、電子メールなどを用いて携帯電話機40の外部からプログラムの実行が指示される場合などもある。

【0070】同図に示すように、まず、携帯電話機40のCPU405は、実行するプログラムとして操作入力により指定されたプログラムを特定する（ステップS101）。次いで、CPU405は、特定したプログラムがダウンロードされたJavaAPであるのか、それともネイティブアプリケーションであるのかを判別する（ステップS102）。前述したようにネイティブアプリケーションには、自身がネイティブアプリケーションであることを示す識別情報が付与されている。したがって、CPU405は、プログラムに上記識別情報が付与されているか否かを判別することで、このプログラムがダウンロードされたJavaAPであるのか、それともネイティブアプリケーションであるのかを判別することができる。

【0071】その結果、CPU405は、プログラムがネイティブアプリケーションであると判別した場合は（ステップS102：No）、オブジェクト生成処理を終了するとともに、実行するプログラムとして指定されたネイティブアプリケーションを起動する。そして、CPU405は、起動させたネイティブアプリケーションに基づく処理を行なう。

【0072】ここで、実行するプログラムがネイティブアプリケーションである場合は、信頼性が完全に保証できるプログラムであるので、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを用いたり、あるいはネイティブアプリケーションの実行に伴ってJAMのアクセス制限機能を動作させる必要がない。したがって、ネイティブアプリケーションが実行される場合、JAMによるアクセス制限は一切行われず、ネイティブアプリケーションは、携帯電話機40内の任意のリソースおよびネットワーク上の任意のリソースにアクセスすることができる。

【0073】一方、CPU405は、プログラムがダウンロードされたJavaAPであると判別した場合は（ステップS102：Yes）、次いで、不揮発性メモリ410に格納されている各種のデータの中から、このJavaAPを実行した場合に使用されるデータを、例えば、このJavaAPのプログラム内容を解析するなどして特定する

(ステップS103)。なお、JavaAPが使用するデータを特定する際には、JARストレージ410b内の、このJavaAPのJARファイルに格納されているデータは特定の対象から除外する。これは、JARファイル内に格納されているデータは、このJavaAPを実行する上で必要となるデータとして当該JavaAPを提供するコンテンツプロバイダが用意したデータであるためである。

【0074】次いで、CPU405は、型指定テーブル410a(図3)を参照して上記特定したデータの型指定情報の値に基づいて、このデータを扱うオブジェクトの型を「完全カプセル化」型とするのか、それとも「非完全カプセル化」型とするのかを決定する(ステップS104)。例えば、JavaAPの使用するデータがアドレス帳データの場合、CPU405は、型指定テーブル410aを参照し、アドレス帳データを扱うオブジェクトの型を「完全カプセル化」型に決定する。また、JavaAPの使用するデータがコンテンツB(著作権保護フラグ“0”)の場合、CPU405は、コンテンツBを扱うオブジェクトの型を「非完全カプセル化」型に決定する。

【0075】この後、CPU405は、上記ステップS103において特定したデータと、上記ステップS104において決定したオブジェクトの型とに基づいて、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成する(ステップS105)。例えば、上記ステップS103において特定したデータがアドレス帳データの場合、CPU405は、オリジナルJava拡張ライブラリ内の完全カプセル化APIを起動して、アドレス帳データ用の完全カプセル化オブジェクトを生成する。また、上記ステップS103において特定したデータがコンテンツBの場合、CPU405は、オリジナルJava拡張ライブラリ内の非完全カプセル化APIを起動して、コンテンツB用の非完全カプセル化オブジェクトを生成する。

【0076】次いで、CPU405は、生成した完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを共通スクラッチパッド410dに格納し(ステップS106)、オブジェクト生成処理を終了する。なお、上記ステップS105において生成された完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、共通スクラッチパッド410dではなく、個別スクラッチパッド410cに格納される形態であってよい。

【0077】また、上記ステップS103においてJavaAPの使用するデータが複数特定された場合は、特定した各データ毎に、当該データ用の完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成して共通スクラッチパッド410dに格納するため、上記ステップS104～S106までの処理を各データ毎に繰り返して行う。そして、CPU405は、オブジェクト

生成処理を終了した後、実行するプログラムとして指定されたJavaAPを起動し、このプログラムに基づく処理を開始する。

【0078】<2. アクセス管理処理>次に、携帯電話機40においてCPU405により実行されるアクセス管理処理について図9を参照して説明する。このアクセス管理処理は、JAMの機能としてCPU405により実行されるものであり、ダウンロードされたJavaAPの実行過程においてアクセス要求が発生した場合に、割り込み処理として実行される。

【0079】同図に示すように、まず、携帯電話機40のCPU405は、JavaAPの実行過程において発生したアクセス要求について、アクセスの要求先が予め許可された範囲内のリソースであるか否かを判別し、アクセス要求を許可するか否かを判定する(ステップS201)。ここで、アクセス要求の許可有無を判定する仕組みについて具体的に説明すると、ダウンロードされたJavaAPが実行される場合、CPU405は、JavaAPの実行に伴ってアクセスすることのできるリソースを、このJavaAPのADFに記述されているURLにより指定される当該JavaAPのダウンロード元のコンテンツサーバ10(サイト)と、このJavaAPに対して割り当てられたJARストレージ410bおよび個別スクラッチパッド410c内の記憶領域と、共通スクラッチパッド410dと、のみに制限する。

【0080】したがって、CPU405は、アクセスの要求先が上述したリソースのいずれかである場合は、このアクセス要求を許可する一方、アクセスの要求先が上述したリソース以外である場合は、このアクセス要求を許可しない。

【0081】次いで、CPU405は、アクセス要求の許可有無を示す判定結果を要求元のJavaAPに通知した後(ステップS202)、アクセス管理処理を終了する。また、実行中のJavaAPは、JAMによる判定結果を受け取ると、この判定結果に従って、アクセス要求が許可された場合は当該アクセス要求に基づく処理を実行する一方、アクセス要求が許可されなかった場合は当該アクセス要求に基づく処理をキャンセルする。

【0082】さて、携帯電話機40のCPU405は、ダウンロードしたJavaAPを実行する場合、図8に示したオブジェクト生成処理を行った後にJavaAPを起動する。また、ダウンロードしたJavaAPの実行過程においてCPU405は、アクセス要求が発生すると、図9に示したアクセス管理処理を行なう。したがって、携帯電話機40は、ダウンロードしたJavaAPの実行中において必ずJAMによるアクセス制限を受けることとなり、例えば、不揮発性メモリ410に格納されているアドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータ、コンテンツなどのデータそのものにアクセスすることができなくなる。

【0083】このため、携帯電話機40のCPU405は、上述したオブジェクト生成処理において、起動させるJavaAPが使用するデータを特定し、当該データ用の完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成して共通スクラッチパッド410dに格納する。この共通スクラッチパッド410dは、前述したように、JAMによるアクセス制限が行われている場合であっても携帯電話機40のアクセスが許可されるリソースである。また、携帯電話機40にダウンロードされるJavaAPは、共通スクラッチパッド410dに格納された完全カプセル化オブジェクトや非完全カプセル化オブジェクトにアクセスし、当該オブジェクトに備わるメソッドを使用してこのオブジェクト内のデータに対する操作を指令するように作成されている。

【0084】例えば、アドレス帳データを使用するJavaAPが起動される場合、上述したオブジェクト生成処理によりアドレス帳データ用の完全カプセル化オブジェクトが生成され、共通スクラッチパッド410dに格納される。また、このJavaAPは、上記生成されたアドレス帳データ用の完全カプセル化オブジェクトに対して、当該オブジェクトに備わるメソッドを用いてこのオブジェクト内のデータに対する操作を指令する。したがって、完全カプセル化オブジェクトの有するアドレス帳データの一部を画面表示させることなどが可能となる一方、完全カプセル化オブジェクトの有するデータそのものがJavaAPに引き渡されることはない。

【0085】従来は、ダウンロードされたJavaAPに対するセキュリティを確保するため、このようなJavaAPについては、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータなどに一切アクセスすることができなかつた。これに対して本実施形態によれば、完全カプセル化オブジェクトを用いることにより、データそのものがJavaAPに引き渡されることがないので、ダウンロードされたJavaAPに対するセキュリティを確保しつつ、同時に、従来は一切アクセスできなかつたデータについて、完全カプセル化オブジェクトを介して画面表示を行わせることができるようになる。したがって、ダウンロードされたJavaAPが携帯電話機40において実現することのできる機能を充実させることができる。

【0086】また、著作権が放棄されたコンテンツや、セキュリティ上の重要度が低く、型指定情報として“0”がセットされているデータを使用するJavaAPが起動される場合、非完全カプセル化オブジェクトが生成されて共通スクラッチパッド410dに格納される。この場合は、完全カプセル化オブジェクトの場合と異なり、非完全カプセル化オブジェクトの有しているデータそのものをJavaAPに引き渡すこともできる。

【0087】すなわち、ダウンロードされたJavaAPは信頼性を完全に保証することのできないプログラムであ

るが、著作権が放棄されたデータやセキュリティ上の重要度が低いデータについては、非完全カプセル化オブジェクトとして扱うことで、JavaAPにデータそのものを引き渡せるようにする。JavaAPにデータそのものを引き渡せるようにした方が利便性を高くできることは説明をするまでもなく明かであり、このようにカプセル化するデータのセキュリティ上の重要度や著作権の保護の要否などに応じて完全カプセル化オブジェクトと非完全カプセル化オブジェクトとを使い分けるようにすると、完全カプセル化オブジェクトのみを用いた場合と比較して、さらに利便性を高めることができる。

【0088】<3. JavaAP終了処理>次に、携帯電話機40においてCPU405により実行されるJavaAP終了処理について図10を参照して説明する。このJavaAP終了処理は、JAMの機能としてCPU405により実行されるものであり、JavaAPの実行終了要求が発生した場合に、割り込み処理として実行される。

【0089】同図に示すように、携帯電話機40のCPU405は、JavaAPの実行終了要求が発生すると、共通スクラッチパッド410dに格納されている完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除する（ステップS301）。このステップS301において削除される完全カプセル化オブジェクトや非完全カプセル化オブジェクトは、JavaAPを起動させる際に、上述したオブジェクト生成処理（図8参照）において生成され、共通スクラッチパッド410dに格納されたものである。CPU405は、共通スクラッチパッド410dからオブジェクトを削除すると、JavaAP終了処理を終える。

【0090】このようにダウンロードされたJavaAPを起動する際に、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを生成して共通スクラッチパッド410dに格納する一方、このJavaAPの実行が終了する際に、共通スクラッチパッド410dから完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除するようになると、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを共通スクラッチパッド410dに常時保持しておく必要がない。したがって、携帯電話機40のメモリ資源を効率的に活用することができる。

【0091】なお、携帯電話機40は、ROM408または不揮発性メモリ410に記憶されているプログラムに従って、本発明に係る処理（オブジェクト生成処理、アクセス管理処理、JavaAP終了処理）を実行するが、このような処理を実行するためのプログラムを携帯電話機40に対して通信により提供する形態としてもよい。さらに、このような処理を実行するためのプログラムを、例えば、光記録媒体や磁気記録媒体、半導体メモリなどの記録媒体を用いて携帯電話機40へ提供するようにしてもよい。但し、プログラムを記録媒体により携帯

電話機40へ提供する場合、携帯電話機40は、記録媒体からプログラムを読み出すための記録媒体ドライブを有する。

【0092】[B. 変形例] 以上、本発明の実施形態について説明したが、本発明はその主要な特徴から逸脱することなく他の様々な形態で実施することが可能である。上述した実施形態は、本発明の一態様を例示したものに過ぎず、本発明の範囲は、特許請求の範囲に示す通りであって、また、特許請求の範囲の均等範囲に属する変形や変更は、全て本発明の範囲内に含まれる。なお、変形例としては、例えば、以下のようなものが考えられる。

【0093】<変形例1> 上述した実施形態では、完全カプセル化オブジェクトと非完全カプセル化オブジェクトとを用いる場合について説明したが、完全カプセル化オブジェクトのみを用いるようにしてもよい。すなわち、ダウンロードされたJavaAPが使用するデータを全て完全カプセル化オブジェクトとして扱うようにしてもよい。この場合、型指定情報は不要となる。また、上述した実施形態では、型指定テーブル410aを用いる構成としたが、アドレス帳データ、電子メールデータ、コンテンツなどのデータ自体に型指定情報を付与するようすれば、指定テーブル410aを用いる必要はない。

【0094】<変形例2> 上述した実施形態では、ダウンロードされたJavaAPの実行が指示された場合に、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトを生成するようにした。しかしながら、完全カプセル化オブジェクトまたは非完全カプセル化オブジェクトが生成されるタイミングは、JavaAPの実行が指示されたときに限定されるものではない。

【0095】 例えば、携帯電話機40の電源投入時などに、型指定テーブル410a(図3)を参照し、各データ用の完全カプセル化オブジェクトや非完全カプセル化オブジェクトを生成して共通スクラッチパッド410dに格納しておくようにしてもよい。但し、この場合、アドレス帳データ、電子メールデータ、着信・発信履歴データ、ユーザデータなどの元データが更新された場合に、その更新内容に応じて共通スクラッチパッド410dに格納されている完全カプセル化オブジェクトや非完全カプセル化オブジェクト内のデータを更新する必要がある。

【0096】 したがって、このような制御を行う場合には、元データの更新に応じてオブジェクト内にカプセル化されているデータを更新するAPIをオリジナル拡張ライブラリに備えるようにする。また、完全カプセル化オブジェクトや非完全カプセル化オブジェクトを共通スクラッチパッド410dに常時保持しておく場合は、当然、JavaAPの実行終了に応じて共通スクラッチパッド410dから完全カプセル化オブジェクトや非完全カプセル化オブジェクトを削除する必要はない。

【0097】<変形例3> 上述した実施形態において、例えば、“1(重要度「高」)”～“5(重要度「低」)”までのセキュリティ上の重要度を示すセキュリティレベル情報をデータに対して付与し、このセキュリティレベル情報が“1”～“3”までのデータを完全カプセル化オブジェクトとして扱う一方、セキュリティレベル情報が“4”および“5”的データを非完全カプセル化オブジェクトとして扱うようにしてもよい。

【0098】<変形例4> 携帯電話機40にダウンロードされたJavaAPであっても、例えば、移動パケット通信網30を運営する通信事業者やCA(Certificate Authority)のような公正な第3者機関によりJavaAPの内容が審査され、一定の動作基準を満たしていると認定されたJavaAPであれば、ネイティブアプリケーションと同様に、信頼性を完全に保証することのできるプログラムとみなすことができる。

【0099】 したがって、第3者機関によって認定されたJavaAPであれば、ダウンロードされたJavaAPであってもネイティブアプリケーションと同様に、使用するデータを不揮発性メモリ410から直接取得できるようにしてもよい。なお、第3者機関により認定されたJavaAPには、当該JavaAPが第3者機関の認定プログラムであることを示す識別情報が付与されている。したがって、携帯電話機40において、ダウンロードされたJavaAPが第3者機関による認定プログラムであるか否かを識別する場合には、上記識別情報の有無を判別すればよい。

【0100】<変形例5> 上述した実施形態においてコンテンツサーバ10は、インターネット20に接続されている構成とした。しかしながら、コンテンツサーバ10は、専用線を介して移動パケット通信網30のゲートウェイサーバ31に直接接続されている構成であってもよい。また、ゲートウェイサーバ31がコンテンツサーバ10の機能を有する構成であってもよい。さらに、コンテンツサーバ10が移動パケット通信網30内に設置されている構成であってもよい。

【0101】<変形例6> 上述した実施形態では、図11においてハッチングで示すように、KVMと、コンフィギュレーションとしてCLDCを備えるとともにプロファイルとしてオリジナルJava拡張プロファイルを備えるJ2MEとが組み込まれた携帯電話機40に本発明を適用した場合について説明した。しかしながら、Java実行環境は、上述したKVMとJ2MEの組み合わせに限定されるものではない。また、本発明が適用可能な通信装置は、携帯電話機に限定されるものではない。

【0102】 例えば、同図に示すように、J2MEのプロファイルとして、オリジナルJava拡張プロファイルの代わりにMIDP(Mobile Information Device Profile)を有する構成であってもよい。また、KVMの代わりにJVMを有し、J2MEのコンフィギュレーションと

してCLDCの代わりにCDC (Connected Device Configuration) を、また、J2MEのプロファイルとして、例えば、液晶付電話機用プロファイル、TV用プロファイル、カーナビゲーション用プロファイルなどを有する構成であってもよい。さらには、HotSpotと、J2SE (Java2 Standard Edition) またはJ2EE (Java 2 Enterprise Edition) とを有する構成であってもよい。

【0103】また、以上説明したJava実行環境の変形例から明らかなように、本発明は、例えば、PHS (Personal Handyphone System:登録商標) 端末やPDA (Personal Digital Assistant)、カーナビゲーション装置、パソコン用コンピュータなどの、通信機能を有する各種電子機器に適用可能である。また、本発明は、移動パケット通信網30に収容される通信装置に限定されるものではない。例えば、図12に示すような通信システム2において、LAN50内に設けられたパソコン用コンピュータ70A~70Cに本発明を適用することもできる。

【0104】また、上述した実施形態では、Javaプログラミング言語により記述されたJavaAPを用いた場合について説明したが、プログラミング言語はJavaに限定されるものではない。

#### 【0105】

【発明の効果】以上説明したように本発明によれば、受信したプログラムに対する通信装置のセキュリティを確保しつつ、このようなプログラムが通信装置において実現することのできる機能を従来と比較して充実させることができる。

#### 【図面の簡単な説明】

【図1】 本発明の実施形態に係る通信システムの構成を例示するブロック図である。

【図2】 同実施形態に係る携帯電話機のハードウェア構成を例示するブロック図である。

【図3】 同実施形態に係る携帯電話機において、不揮発性メモリに格納されている型指定テーブルのデータ構成を例示する図である。

【図4】 同実施形態に係る携帯電話機において、JavaAPの実行環境を説明するための図である。

【図5】 同実施形態に係る携帯電話機において、カプセル化オブジェクトを説明するための模式図である。

【図6】 同実施形態に係る携帯電話機において、非完全カプセル化オブジェクトについて例示する模式図である。

【図7】 同実施形態に係る携帯電話機において、完全カプセル化オブジェクトについて例示する模式図である。

【図8】 同実施形態に係る携帯電話機において、CPUにより実行されるオブジェクト生成処理の動作を説明するフローチャートである。

【図9】 同実施形態に係る携帯電話機において、CPUにより実行されるアクセス管理処理の動作を説明するフローチャートである。

【図10】 同実施形態に係る携帯電話機において、CPUにより実行されるJavaAP終了処理の動作を説明するフローチャートである。

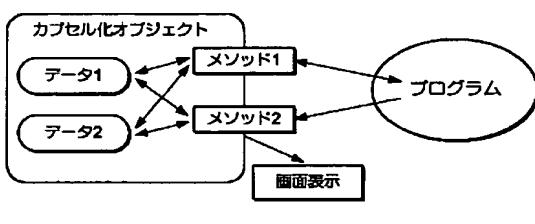
【図11】 本発明の変形例6に係り、Java実行環境の変形例を説明するための図である。

【図12】 本発明の変形例6に係り、通信システムの変形例を例示するブロック図である。

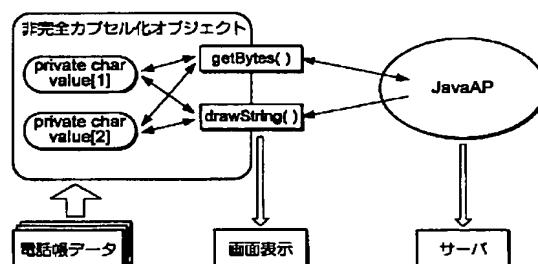
#### 【符号の説明】

1, 2……通信システム、10……コンテンツサーバ、20……インターネット、30……移動パケット通信網、31……ゲートウェイサーバ、32……基地局、40……携帯電話機、50……LAN、60……ゲートウェイサーバ、70A, 70B, 70C……パソコン用コンピュータ、401……無線通信部、401a……アンテナ、402……操作入力部、403……通話処理部、404……通信インターフェース、405……CPU、406……液晶表示部、407……記憶部、408……ROM、409……RAM、410……不揮発性メモリ、410a……型指定テーブル、410b……JARストレージ、410c……個別スクラッチパッド、410d……共通スクラッチパッド、411……バス。

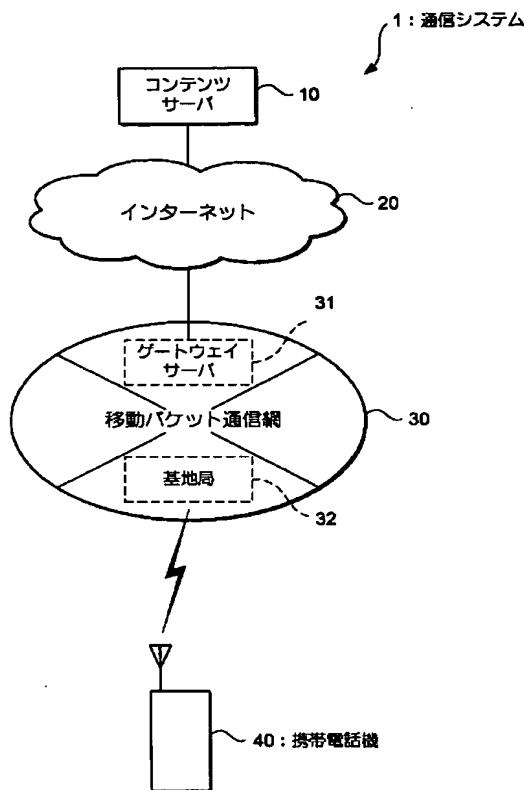
【図5】



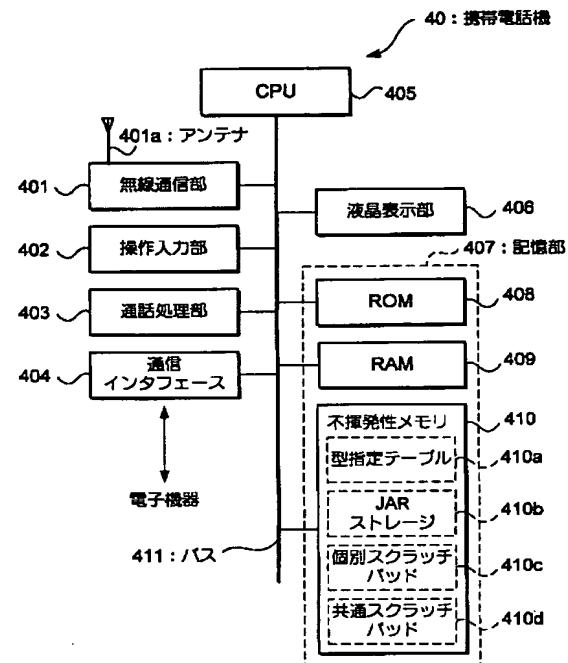
【図6】



【図 1】



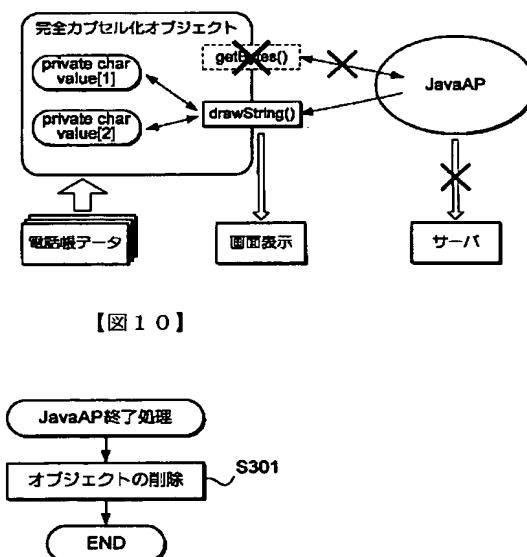
【図 2】



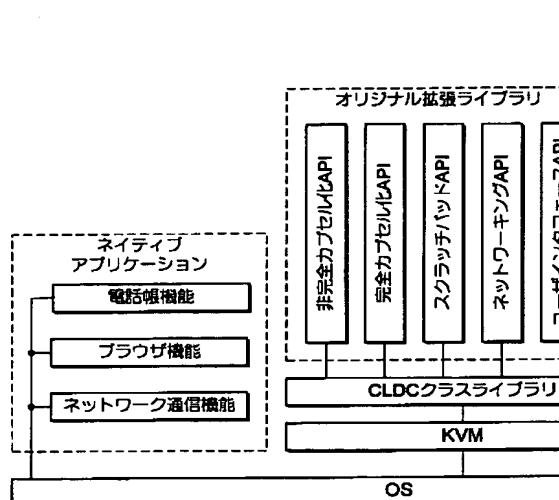
【図 3】

410a: 型指定テーブル	
データ名	型指定情報
アドレス帳データ	1 (完全カプセル化型)
電子メールデータ	1
着信・発信履歴データ	1
ユーザデータ	1
コンテンツA (著作権保護フラグ "1" )	1
コンテンツB (著作権保護フラグ "0" ) (非完全カプセル化型)	0
自作画像データ	0
⋮	⋮

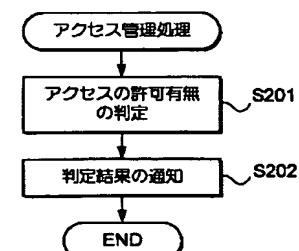
【図 7】



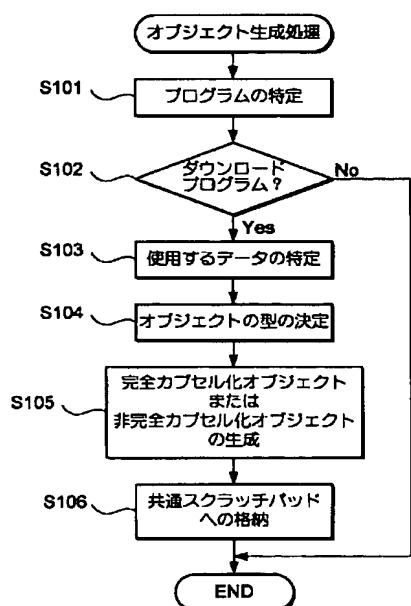
【図 4】



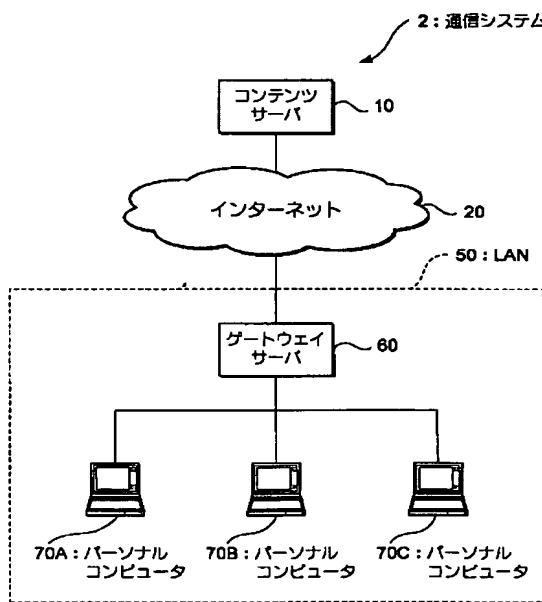
【図 9】



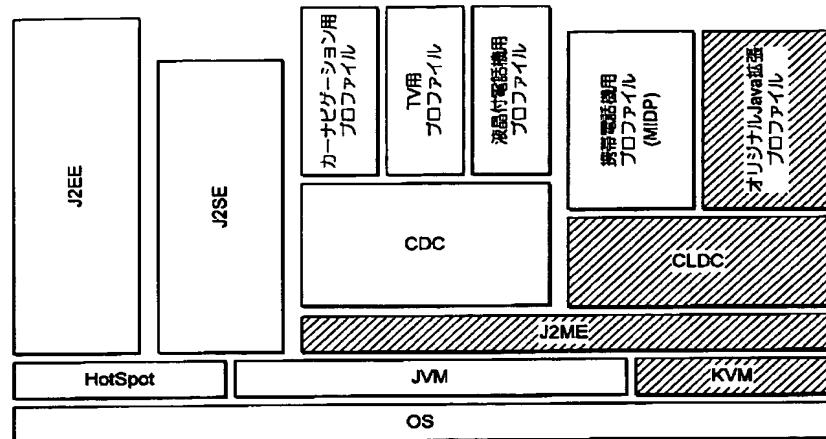
【図 8】



【図 12】



【図 1 1】



---

フロントページの続き

(72) 発明者 山田 和宏  
東京都千代田区永田町二丁目11番1号 株  
式会社エヌ・ティ・ティ・ドコモ内

(72) 発明者 鶩見 豊  
東京都港区虎ノ門4-3-20 神谷町MT  
ビル2F 株式会社ケイ・ラボラトリ一内  
F ターム(参考) 5B017 AA03 AA07 BA06 BA08 CA16  
5K067 AA30 AA32 BB04 CC08 DD57  
EE03 EE10 EE16 EE23 EE35  
EE37 FF07 GG11